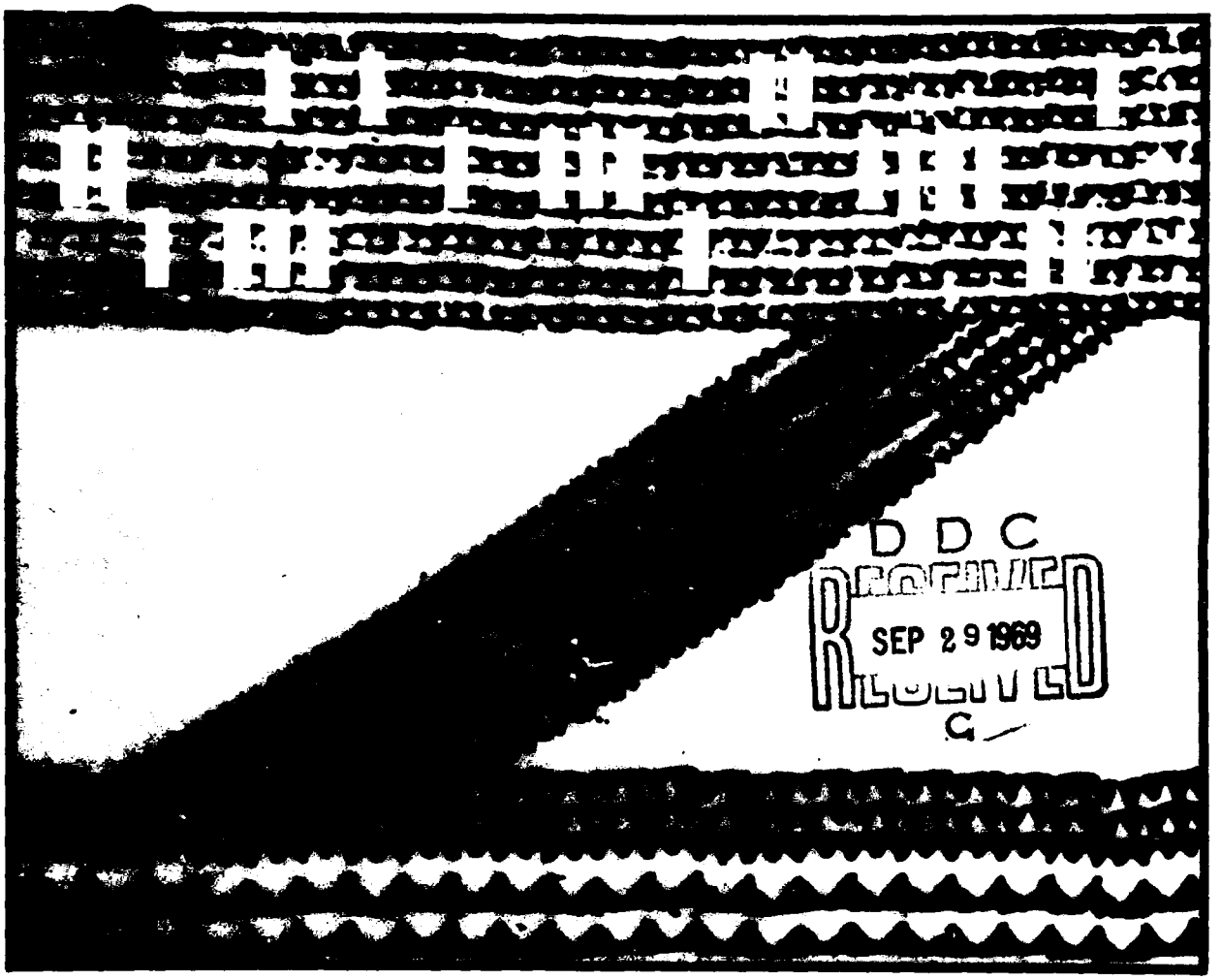
AFOSR 69-2027TR

693622

COMPUTER SCIENCES VS SOFTWARE ENGINEERING

by

FRANKLIN F. KUO



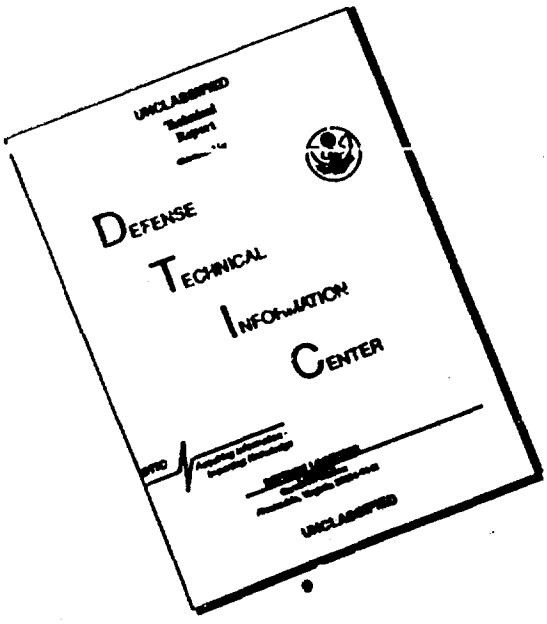
THE SYSTEM

1. This document has been approved for public release and sale; its distribution is unlimited.

July 1969
Report A69-4

7

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

COMPUTER SCIENCES VS SOFTWARE ENGINEERING

ABSTRACT

In this editorial written for COMPUTER DECISIONS, we advance the thesis that computer science departments in the U.S. are not fulfilling the needs of the software industry. We propose a new program called "software engineering" to fill the void. Software engineering stresses the pragmatic side of computer systems design and could be established as an independent program with either existing electrical engineering or computer sciences departments.

Computer Sciences vs Software Engineering†

by

Franklin F. Kuo
Professor of Electrical Engineering and
Information Sciences
University of Hawaii

There are about 100 computer science departments, most of them offering graduate degrees, in American universities today. Many are interdisciplinary in structure. However, the "core subjects" offered by these departments are in the computer sciences. Without attempting to define precisely what computer science is, we can list a number of core courses common to most of the departments' curricula. These courses include: automata theory, formal languages and syntactic analysis, artificial intelligence and heuristic programming, theory of computability, etc. For a complete list, the reader should refer to the report of the ACM Curriculum Committee on Computer Science, CURRICULUM 68*.

With all of the above courses, one would think that the computer science graduate of 1969 would be well equipped to tackle the intricacies associated with third or fourth generation computer systems - both in hardware design and software implementation. However,

---

† This work is supported by THE ALOHA SYSTEM, a research project of the University of Hawaii, supported by the Office of Aerospace Research (SRMA) under contract number F44620-69-C-0030, a Project THEMIS award.

*"Curriculum 68", Communications of the ACM, Vol. 11, No. 3, March 1968, pp. 151-197.

this is hardly the case. The reason is that most computer science departments do not emphasize enough the practical aspects of computer systems design. Many computer science departments reside either in the graduate school or in the college of liberal arts and sciences. Therefore, it is not inconsistent for these departments to emphasize the "science" aspects of computers and computing. However, we must realize that computer sciences in the academic sense have very little to do with real computers. The relationship between the abstract and applied sides of computer sciences is increasingly similar to that which exists between pure and applied mathematics.

Computer sciences is becoming more involuted. Theorems are generated which depend only upon previous theorems. There is beauty in them for the scientist - but there is (usually) little utility. For the academician an abstract approach to computer sciences also serves in writing technical papers - with the accompanying prestige and fame. On the other hand there is little publications mileage to be gained in designing and implementing a special software package. The time required to implement this package might be two or three times that of doing the research and writing a paper in a field such as automata theory. In most universities, promotions and other tangible rewards come as the result of publications. An efficient compiler is hardly regarded as a respectable publication.

The computer scientists claim that they are seeking to understand the fundamental nature of computers and that what they are doing will ultimately benefit computing, society and mankind. I

see little evidence of these benefits, but perhaps I haven't held my breath long enough. This standard argument, however, cannot be refuted by counter-examples. If we recognize the trend towards abstractness that prevails in most computer science departments and if we view this trend as irreversible - then we must decide on an alternative means of training the computer systems people that industry so desperately needs.

My suggestion is to establish a curriculum called software engineering (or something equivalent as computer systems engineering, or have an option in the standard computer science curriculum with an asterisk (*) denoting practical). This suggestion, by no means original, is based upon the widely held opinion that a computer system is an integrated entity involving both hardware and software. To design effective software, one must have a real understanding of the hardware capabilities of the system. Therefore the approach we must adopt to write efficient software is engineering methodology. In the software engineering curricula, courses such as machine organization, compiler construction, systems programming, system simulation and switching theory will be given considerable emphasis. I realize that not many software engineering curricula exist today. However, they can readily be established by an amalgamation of electrical engineering and computer science departments and exist as a separate program within either or both departments. The need for software engineers is critical. It is time to stop kidding ourselves, computer science departments as a whole, are simply not meeting this need.

Finally, I am reminded of a statement a friend of mine made 11 years ago when the Russians had just put up their Nth sputnik. "Don't worry," he said, "we'll catch up in space for sure. The Russians are becoming interested in information theory." The situation is reversed today when we look at computers in the U.S. and the Soviet Union. I have a feeling that the Russians will catch up with the U.S. in computer power because some of our best men have gotten into computer sciences.

Security Classification

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified

1. ORIGINATING ACTIVITY (Corporate author) University of Hawaii Department of Electrical Engineering Honolulu, Hawaii 96822	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

COMPUTER SCIENCES VS SOFTWARE ENGINEERING

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Interim

5. AUTHOR(S) (First name, middle initial, last name)

Franklin F. Kuo

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 1969	6	1

8a. CONTRACT OR GRANT NO	9a. ORIGINATOR'S REPORT NUMBER(S)
F44620-69-C-0030	
b. PROJECT NO 9749-01	
c. 6144501F	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFOSR 69-2027TR
d. 681304	

10. DISTRIBUTION STATEMENT

1. This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
TECH, OTHER	Air Force Office of Scientific Research (SRMA) 1400 Wilson Boulevard Arlington, Virginia 22209

13. ABSTRACT

~~In this~~ editorial written for COMPUTER DECISIONS, ~~we~~ advance the thesis that computer science departments in the U. S. are not fulfilling the needs of the software industry. ~~We propose~~ a new program called "software engineering" to fill the void. Software engineering stresses the pragmatic side of computer systems design and could be established as an independent program with either existing electrical engineering or computer sciences departments.

DD FORM 1473